
Интеграция PHP-проекта с 1С
Пётр Мязин
php
PHP Russia
2022
1

О докладчике: Пётр Мязин

- Работаю в PHP-экосистеме более 10 лет
- Веду подкаст Пятиминутка PHP
  5minphp.ru
- Создаю системы учёта для логистических компаний
- Работаю в Групе компаний Forward — современный цифровой экспедитор
  group-forward.ru





ЦИФРОВИЗАЦИЯ СЕРВИСОВ FORWARD GROUP







ЕДИНЫЙ КЛИЕНТСКИЙ ПОРТАЛ

Грузовой и финансовый документооборот, оформление договоров, прием обращений клиентов и обратная связь.



ОНЛАЙН-СЕРВИСЫ

Специализированные платформы для экспедиторов и грузоперевозчиков.



ЭЛЕКТРОННЫЙ ДОКУМЕНТООБОРОТ

Погрузочные документы, инструкции по обработке, отчеты, закрывающие документы, УПД.



МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

Единое информационное пространство для клиентов, оперативная информация о движении груза.

ИНТЕГРАЦИЯ В КОРПОРАТИВНЫЕ СИСТЕМЫ

- ✓ Сервис по IT-интеграции с корпоративной ERP-системой и юридически значимый электронный документооборот (счета-фактуры, акты) через системы СБИС и Диадок
- ✓ Передача данных в форматах XML, JSON, EDIFACT, Excel и других, через веб-сервисы (GraphQL, REST API, SOAP), sftp, email
- ✓ Подготовка Excel-отчётов в требуемом формате с автоматической доставкой на почту по расписанию



О чём доклад?

- Об интеграции PHP-проекта с 1С
- Делюсь личным опытом
- Покажу интересные примеры
- Расширение кругозора





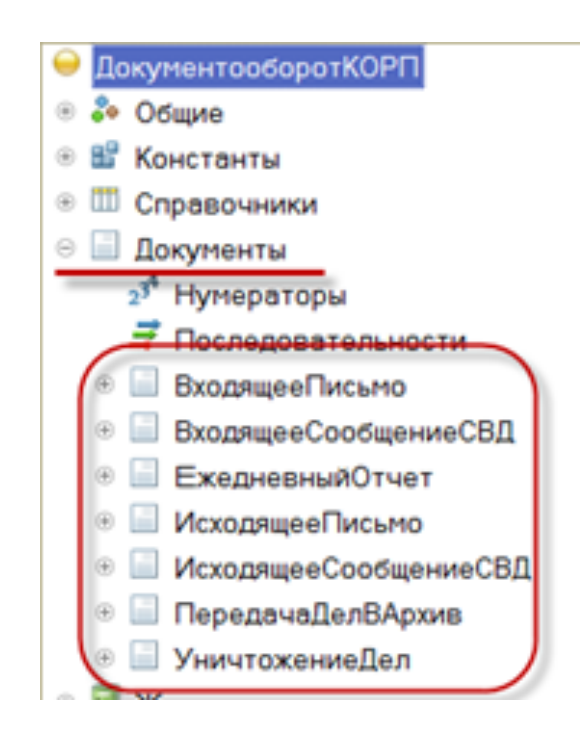
- ~~Битрикс~~
- ~~Интернет-магазины, выгрузка товаров~~
- ~~Commerce ML~~
- ~~Enterprise Data~~



Объекты метаданных в 1С

- **Справочники**
  Примеры: клиенты, города, список товаров или услуг
- **Документы**
  Примеры: счета, акты, заказы, письма, события реального мира
- **Регистры**
  Исторические данные (курсы валют, цены на определённую дату), приход/уход по складам

- Много чего ещё: бизнес-процессы, обработки, отчёты, константы, роли, регламентные задания…

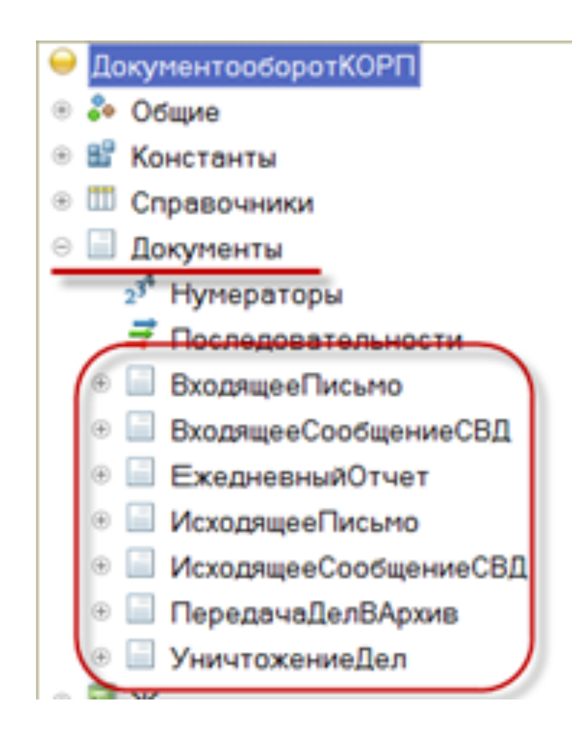




На что похоже?



- Справочники **= Entity / Модели**
  Примеры: клиенты, города, список товаров или услуг
- Документы **= Entity / Модели**
  Примеры: счета, акты, заказы, письма, события реального мира
- Регистры **= История событий / Event Sourcing / Кэш**
  Исторические данные (курсы валют, цены на определённую дату), приход/уход по складам



- Много чего ещё: бизнес-процессы, обработки, отчёты, константы, роли, регламентные задания…



Нужно ли синхронизировать?

- Имеем одинаковые объекты учёта в двух системах: 1С и PHP
- Пользователи работают с веб-интерфейсом (PHP-приложением), но нужны данные из 1С
- Синхронизируем только то, что реально нужно

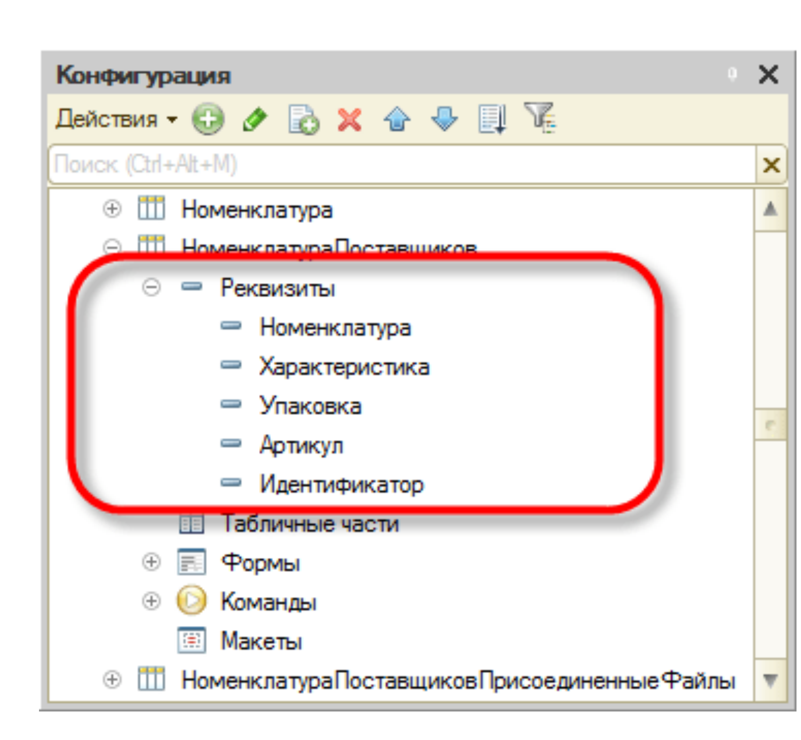






Что синхронизируем?

- **Контрагенты** (клиенты, подрядчики)
- **Договора** с контрагентами
- **Номенклатура** (товары и услуги)
- **Счета, акты, оплаты**



Что НЕ синхронизируем?

- Акты сверки — этим занимается чисто бухгалтерия
- Регламентированные отчёты (в налоговую и прочие органы)
- Приказы о приёме на работу, переводы, увольнения (1С ЗУП)



1 1С:Шина	O Обмен данными	R REST интерфейс
B Встроенный веб-кли-ент	H HTTP-сервисы	J JSON
W Web-сервисы	X XML-документы	X XDTO
P Работа с HTTP и FTP	P Работа с электронной почтой	T Технология внешних компонентов
B Внешнее соединение	P Работа с файлами	H HTML-документы
T Текстовые документы	T Текстовые файлы	A Automation Client/Server



Способы передачи данных



- COM, ODBC
- Обмен файлами: XML, CSV
- HTTP сервисы, REST



Кто инициатор обмена?
Кто источник правды?



- PHP приложение
- 1С
- Двунаправленная синхронизация, используем UUID









PHP-клиент 👉 1С-сервер

REST



1C REST

- 1С **автоматически генерирует REST** интерфейс ко всем прикладным данным
- **Получение списка** документов, справочников, записей регистра
- **Получение элемента** справочника, документа
- **Редактирование** элемента справочника, документа
- **Создание** элемента справочника, документа, набора записей





Это **классический** REST

- **GET** — получение данных
- **POST** — создание объекта
- Обновление данных:
    - **PATCH** — указываем только те свойства, которые обновляем
    - **PUT** — указываем все свойства сущности
- **DELETE** — удаление объекта



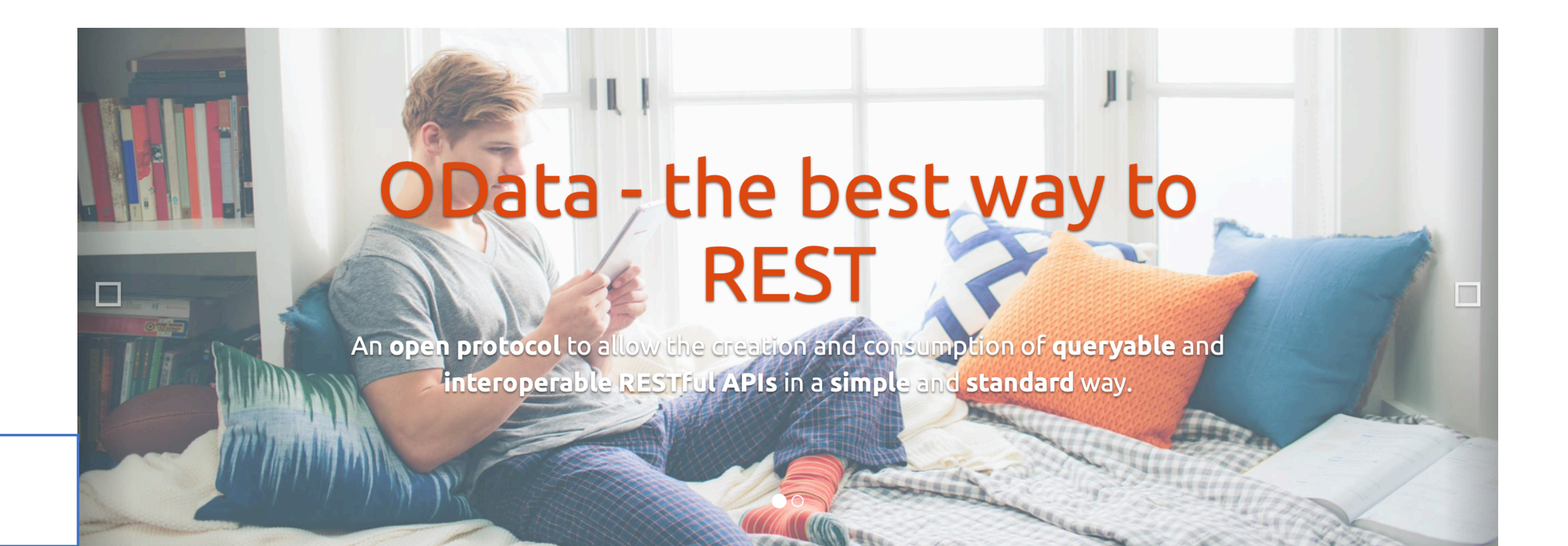
1C REST – выполнение **команд**

- Для документа — проведение и отмена проведения
- Для задачи — выполнение
- Для бизнес-процесса — старт
- Для регистра сведений — получение среза первых и среза последних
- Для регистра накопления и регистра бухгалтерии — получение остатков, оборотов, остатков и оборотов



OData 3.0

- https://www.odata.org
- An open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs in a simple and standard way
- ISO/IEC approved, OASIS standard





Open Data Protocol (OData)

- https://en.wikipedia.org/wiki/Open_Data_Protocol
- Инициатива Microsoft 2007 года
- Открытый веб-протокол для запроса и обновления данных
- Данные в формате **XML** или **JSON**
- Есть библиотеки на всех популярных языках
- В том числе для **PHP**
- saintsystems/odata-client-php
- В том числе обёртки для **Yii2** и **Laravel** (не пробовал)



OData на **Хабре**

- PHP и OData: пересаживаемся с велосипедов на технологию от Microsoft https://habr.com/ru/post/267811/

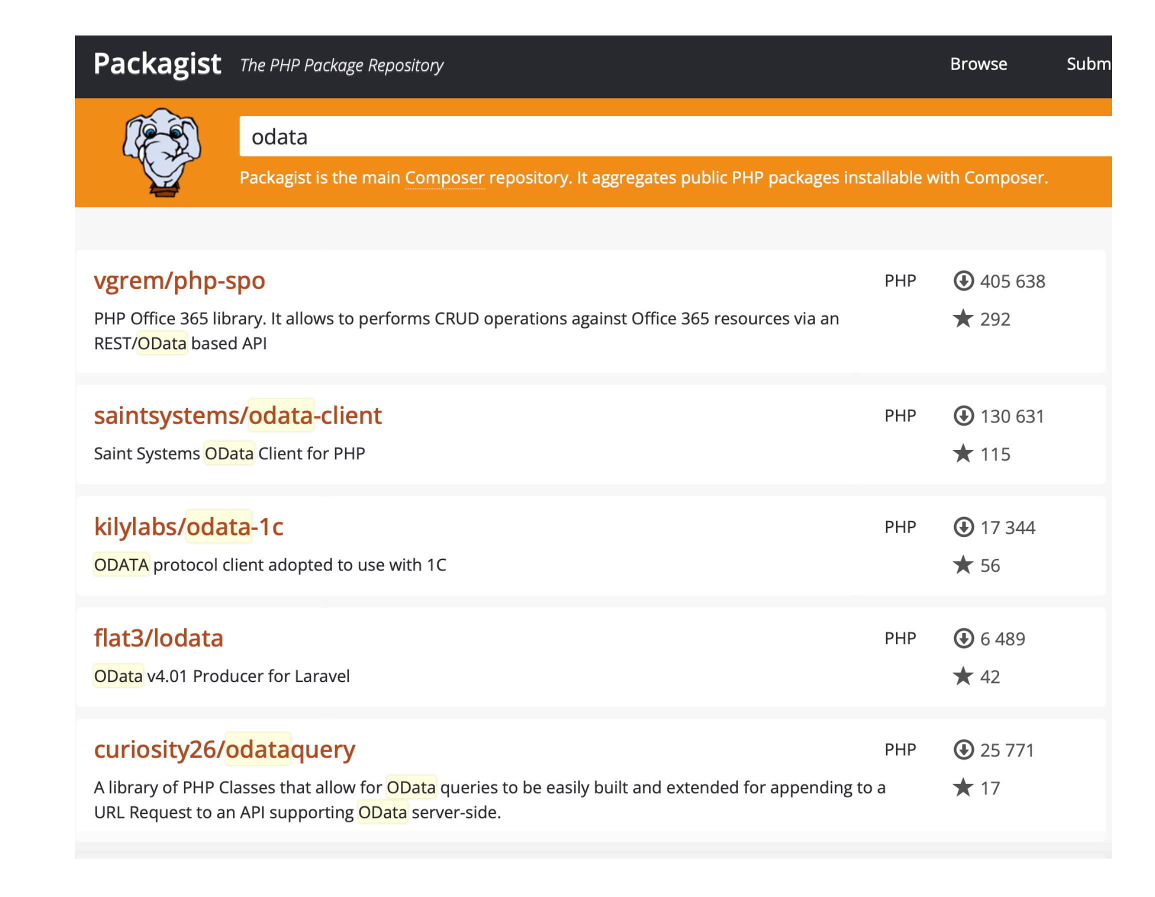
- Кратко об OData https://habr.com/ru/post/678614/



OData на **Хабре**

- PHP и OData: пересаживаемся с велосипедов на технологию от Microsoft https://habr.com/ru/post/267811/
- Кратко об OData https://habr.com/ru/post/678614/

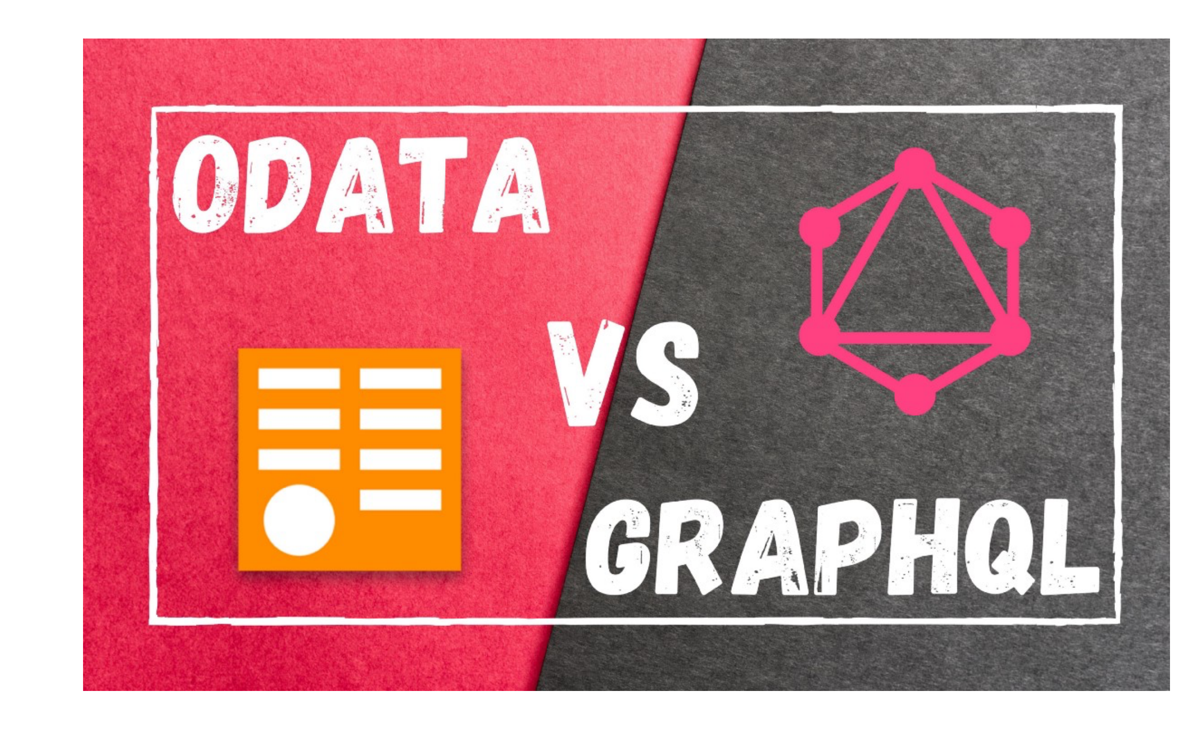
OData на **Packagist**





OData — **GraphQL** на минималках

- Выбор полей для чтения
- Можно выбрать связанные/вложенные объекты (~~N+1~~)
- Фильтры, отборы и условия в запросе
- Google it: **OData vs GraphQL**





Примеры запросов к 1С REST

- Выбрать товары «Молоко» с ценой менее 25 руб

```
GET /Catalog_Товары?$filter=Имя eq 'Молоко' and Цена lt 2500
```

- Получить справочник контрагентов, выбрать только ключ и описание

```
GET /Catalog_Контрагенты?$select=Ref_Key,Description
```




GET /Catalog_Контрагенты?**$select=Ref_Key,Description**&format=json


```
{
  "odata.metadata": "http://192.168.20.251:8002/ForwardTransBuh/odata/standard.odata/
$metadata#Catalog_Контрагенты",
  "value": [{
    "Ref_Key": "1fb98071-fa56-11e5-80bf-005056a30667",
    "Description": "Велоснабжение"
  },{
    "Ref_Key": "64fd5f7e-f58b-11e5-80bf-005056a30667",
    "Description": "ДЕЛФИН ДИСТРИБЬЮШЕН"
  },{
    "Ref_Key": "579a1df3-fbfb-11e5-80bf-005056a30667",
    "Description": "Диамант"
  },{
    "Ref_Key": "c5e0c629-fcba-11e5-80bf-005056a30667",
    "Description": "ДС Контролз"
  },{
    "Ref_Key": "8ddd94f8-fcdf-11e5-80bf-005056a30667",
    "Description": "ИП Чевелюк Александр Васильевич"
  },{
```

Query-параметры запроса на чтение

- **$filter** — отбор при получении данных
- **$select** — перечисление свойств сущности, которые попадут в результат запроса
- **$top** — ограничение количества возвращаемых записей
- **$skip** — убирает из результата запроса указанное количество
- **$count** — возвращает количество записей
- **$inlinecount=allpage(=none)** — добавляет в результат информацию о количестве записей
- **$orderby**=<Реквизит1> asc, <Реквизит2> desc — сортировка



Примеры операторов

- Логические операции:

Описание	Имя	Пример
Равно	eq	/Catalog_Города?$filter=Description eq 'Главный'
Не равно	ne	/Catalog_Города?$filter=Description ne 'Пермь'
Больше	gt	/Catalog_Товары?$filter=Цена gt 10
Больше или равно	ge	/Catalog_Товары?$filter=Цена ge 10
Меньше	lt	/Catalog_Товары?$filter=Цена lt 10
Меньше или равно	le	/Catalog_Товары?$filter=Цена le 10
Логическое ИЛИ	or	/Catalog_Товары?$filter=Цена lt 10 or Цена gt 100
Логическое И	and	/Catalog_Товары?$filter=Цена gt 10 and Цена lt 100
трицание	not	/Catalog_Товары?$filter=not (Цена eq 10)



Примеры операторов

- Логические операции:

Описа
Равно
Не рав
Больш
Больш
Меньш
Меньш
Логич
Логич
триц

- Арифметические операции:

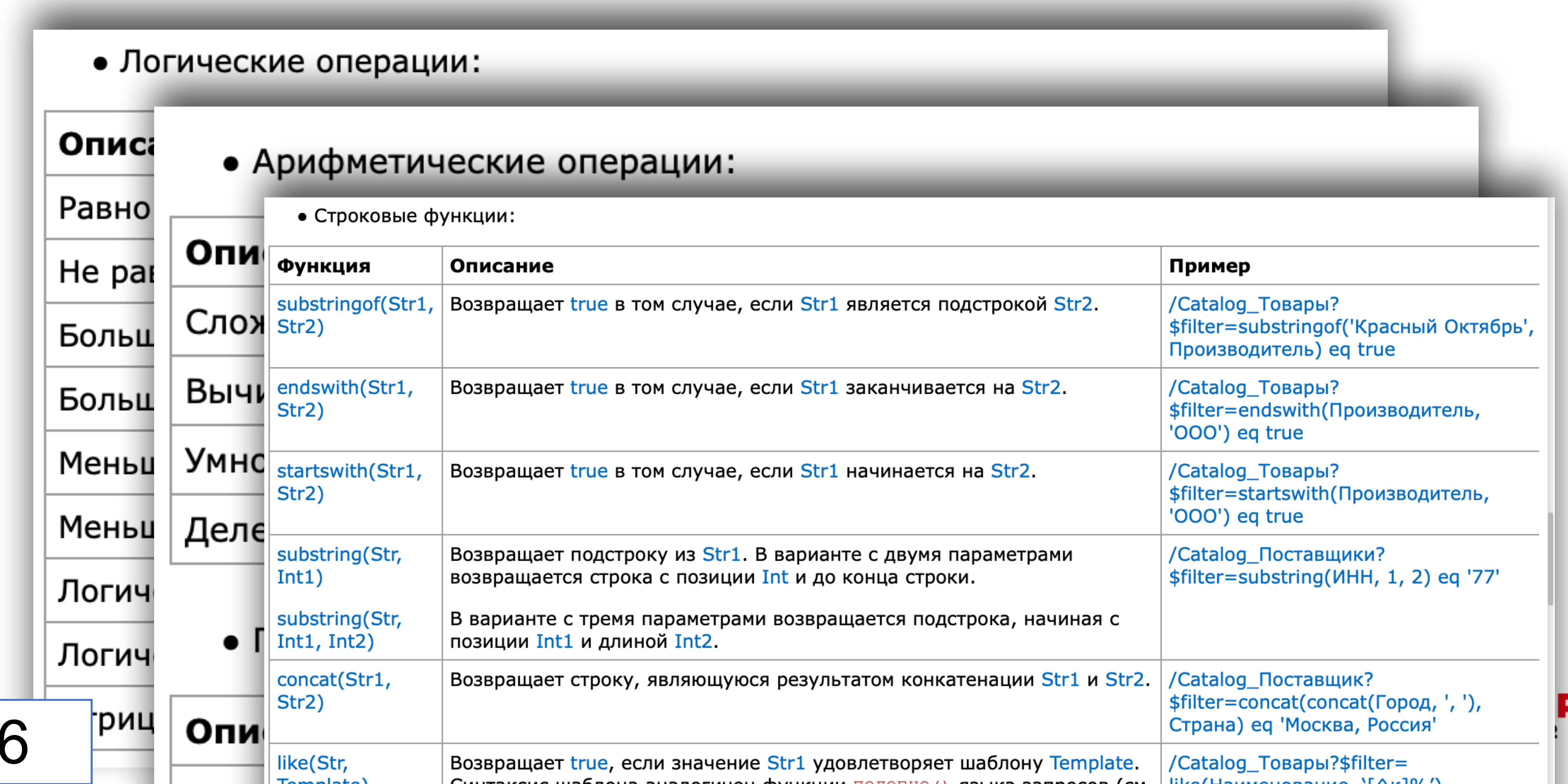
Описание	Имя	Пример
Сложение	add	/Catalog_Товары?$filter=Цена add 5 gt 10
Вычитание	sub	/Catalog_Товары?$filter=Цена sub 5 gt 10
Умножение	mul	/Catalog_Товары?$filter=Цена mul 5 gt 1000
Деление	div	/Catalog_Товары?$filter=Цена div 4 gt 2

- Группирующие операторы:

Описание	Имя	Пример



Примеры операторов



• Логические операции:

• Арифметические операции:

• Строковые функции:

Функция	Описание	Пример
substringof(Str1, Str2)	Возвращает true в том случае, если Str1 является подстрокой Str2.	/Catalog_Товары?$filter=substringof('Красный Октябрь', Производитель) eq true
endswith(Str1, Str2)	Возвращает true в том случае, если Str1 заканчивается на Str2.	/Catalog_Товары?$filter=endswith(Производитель, 'ООО') eq true
startswith(Str1, Str2)	Возвращает true в том случае, если Str1 начинается на Str2.	/Catalog_Товары?$filter=startswith(Производитель, 'ООО') eq true
substring(Str, Int1) substring(Str, Int1, Int2)	Возвращает подстроку из Str1. В варианте с двумя параметрами возвращается строка с позиции Int и до конца строки. В варианте с тремя параметрами возвращается подстрока, начиная с позиции Int1 и длиной Int2.	/Catalog_Поставщики?$filter=substring(ИНН, 1, 2) eq '77'
concat(Str1, Str2)	Возвращает строку, являющуюся результатом конкатенации Str1 и Str2.	/Catalog_Поставщик?$filter=concat(concat(Город, ', '), Страна) eq 'Москва, Россия'
like(Str,	Возвращает true, если значение Str1 удовлетворяет шаблону Template.	/Catalog_Товары?$filter=

$expand

- Позволяет вместе с результатами основного запроса получать значения связанных сущностей
- Позволит не запрашивать каждую сущность отдельно
- Решает проблему N + 1 запроса

```
GET /Catalog_Контрагенты?$expand=ОсновнойДоговорКонтрагента
```

```
GET /Catalog_Контрагенты?$expand=*
```




```
GET /Catalog_Контрагенты?
$select=Ref_Key,Description,ОсновнойДоговорКонтрагента
&$expand=ОсновнойДоговорКонтрагента&format=json
```


```
{
  "odata.metadata": «https://server1c/ForwardTransBuh/odata/standard.odata/
$metadata#Catalog_Контрагенты",
  "value": [
    {
      "Ref_Key": "93efe5be-f74e-11e5-80bf-005056a30667",
      "Description": "АВТОДОР Дата прекращения деятельности: 26.03.2020",
      "ОсновнойДоговорКонтрагента@navigationLinkUrl":
"Catalog_Контрагенты(guid'93efe5be-f74e-11e5-80bf-005056a30667')/
ОсновнойДоговорКонтрагента",
      "ОсновнойДоговорКонтрагента": {
        "Owner_Key": "93efe5be-f74e-11e5-80bf-005056a30667",
        "Parent_Key": "00000000-0000-0000-0000-000000000000",
        "ВалютаВзаиморасчетов_Key": "5ac3d39e-683d-11e0-8eb3-000ffe417543",
        "Организация_Key": "17bd2197-fc1a-11dc-81b5-001d9223bf33",
        "ТипЦен_Key": "00000000-0000-0000-0000-000000000000",
        "ВидВзаиморасчетов_Key": "00000000-0000-0000-0000-000000000000",
```

```
GET /Catalog_Контрагенты?
$select=Ref_Key,Description,ОсновнойДоговорКонтрагента/
Номер,ОсновнойДоговорКонтрагента/СрокДействия
&$expand=ОсновнойДоговорКонтрагента&format=json
```

```
{
  "odata.metadata": «https://server1c/ForwardTransBuh/odata/standard.odata/
$metadata#Catalog_Контрагенты",
  "value": [
    {
      "Ref_Key": "93efe5be-f74e-11e5-80bf-005056a30667",
      "Description": "АВТОДОР Дата прекращения деятельности: 26.03.2020",
      "ОсновнойДоговорКонтрагента@navigationLinkUrl":
"Catalog_Контрагенты(guid'93efe5be-f74e-11e5-80bf-005056a30667')/
ОсновнойДоговорКонтрагента",
      "ОсновнойДоговорКонтрагента": {
        "Номер": "13",
        "СрокДействия": "0001-01-01T00:00:00"
      }
    },
    {
```




/Catalog_Контрагенты(**guid**’88d54406-36a1-11e9-8bb2-642737df2048')?format=json

```
{
  "odata.metadata": «https://server1c/ForwardTransBuh/odata/standard.odata/
$metadata#Catalog_Контрагенты/@Element",
  "Ref_Key": "1fb98071-fa56-11e5-80bf-005056a30667",
  "Description": "Велоснабжение",
  "КодПоОКПО": "",
  "Code": "000000642",
  "ГоловнойКонтрагент_Key": "1fb98071-fa56-11e5-80bf-005056a30667",
  "ОсновнойДоговорКонтрагента_Key": "1fb98072-fa56-11e5-80bf-005056a30667",
  "Комментарий": "",
  "НаименованиеПолное": "ООО \"Велоснабжение\"",
  "КПП": "502701001",
  "Parent_Key": "00000000-0000-0000-0000-000000000000",
  "Predefined": false,
  "ИНН": "5027210205",
  "DataVersion": "AAAAAABtfK4=",
  "IsFolder": false,
```

POST / PUT / PATCH **на запись в 1С**

```
PATCH /Document_СчетНаОплатуПокупателю(guid'3dfa2534-2abe-11ed-84b6-b94427f761
```

- В теле сообщения **JSON-**структура
- Используем **UUID** как ключ (УникальныйИдентификатор)
- В документах могут быть **табличные части** — передаются как вложенные массивы

```
{
  "Шапка": "Какая-то шапка",
  "Товары": [
   {
     "LineNumber": "1",
     "Номенклатура_Key": "c08d547a-0ca0-11ec-8db9-
     "Содержание": "Вознаграждение",
     "Количество": 2,
     "Цена": 100,
     "Сумма": 200,
     "ПроцентСкидки": 0,
     "СуммаСкидки": 0,
     "СтавкаНДС": "НДС0",
     "СуммаНДС": 0
   }
  ]
}
```

Запись: **оптимистичные блокировки**

Проверка, что данные **не изменились с момента считывания**

Заголовок `If-Match` HTTP-запроса PATCH или DELETE

В качестве значения заголовка должно выступать значение свойства `DataVersion`, которое получено при предварительном чтении сущности



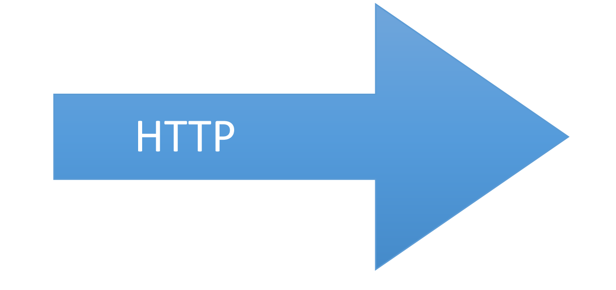
Выводы про 1С REST и OData

- Гибко
- Удобно
- Просто
- В простейшем случае и библиотеки никакие не нужны!

```
file_get_contents('https://server1c/Catalog_Контрагенты?$expand=*');
```







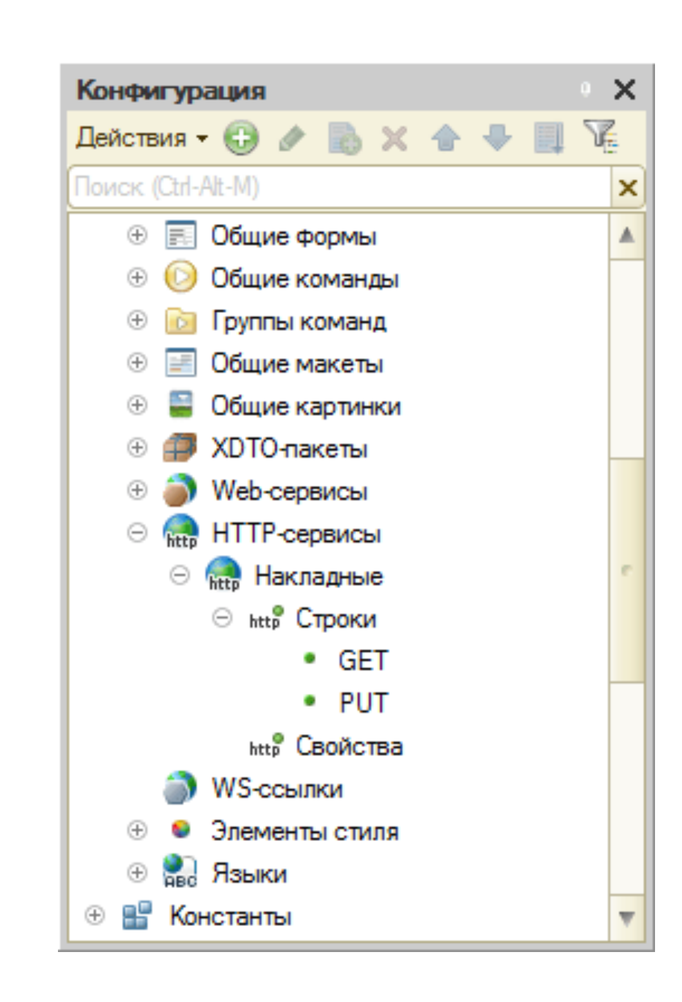
PHP-клиент 👉 1C-сервер

HTTP (не ~~REST~~)



HTTP-сервисы в 1С

- Это «**контроллеры**» на языке 1С
- Функция на 1С получает входящий HTTP-запрос
- Разбираем параметры и тело запроса
- Выполняем какую-то полезную работу
- Возвращаем HTTP-ответ





HTTP-сервисы вместо REST (OData)?

- **Генерация PDF- и Excel-**документов (УПД, счет-фактура)
- Отправка документов по **ЭДО**
- **Запуск** особых **процедур**, например «закрытие заказа» (внутри логика на 1С)



Роутер на 1С

HTTP-сервис Накладные

Основные
Подсистемы
ШаблоныURL
Прочее

Строки
- GET
- PUT

Свойства

Действия | <Назад | Далее> | Закрыть | Справка

Свойства: Строки

Поиск (Ctrl-Alt-I)

Основные:

Имя	Строки
Синоним	Строки
Комментарий	
Шаблон	/{Номер}/Строки/{НомерСтроки}

Свойства: GET

Поиск (Ctrl-Alt-I)

Основные:

Имя	GET
Синоним	GET
Комментарий	
HTTP-метод	GET
Обработчик	СтрокиGET



37

HTTP-сервис Накладные: Модуль

```
// Функция возвращает данные N-ой строки расходной накладной.
// Номер нужной накладной и номер строки содержатся в полученном HTTP запросе.
//
// Параметры:
//  Запрос - HTTPСервисЗапрос
// Возвращаемое значение:
//    - HTTPСервисОтвет
Функция СтрокиGET(Запрос)

    // Пример запроса:
    //      HTTP://test.server.ru/hs/Накладные.hs/000000012/Строки/1

    // Разобрать URL запроса.
    ЗапросНомерДокумента = Запрос.ПараметрыURL["Номер"];
    ЗапросНомерСтроки    = Запрос.ПараметрыURL["НомерСтроки"];

    НужныйДокумент = Документы.РасходТовара.НайтиПоНомеру(ЗапросНомерДокумента);

    // Обработка ошибочных ситуаций: документ не найден, номер не задан.
    Если НужныйДокумент = Неопределено ИЛИ НужныйДокумент.Пустая() Тогда
        Ответ = Новый HTTPСервисОтвет(404);

        Возврат Ответ;

    КонецЕсли;

    // Вернуть данные строки.
    СтрокаДокумента = НужныйДокумент.Товары[Число(ЗапросНомерСтроки) - 1];

    // Преобразовать данные строки в XML.
    ЗаписьXML = Новый ЗаписьXML;
    ...сьXML.УстановитьСтроку();
    ...сьXML.ЗаписатьОбъявлениеXML();
    ...сьXML.ЗаписатьНачалоЭлемента("answer");
```

Пример функции обработчика входящего HTTP-запроса

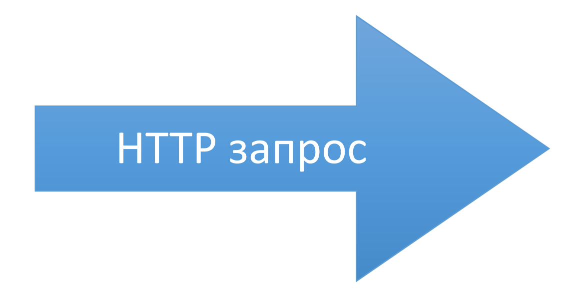


```
	Если НужныйДокумент = Неопределено ИЛИ НужныйДокумент.Пустая() Тогда
		Ответ = Новый HTTPСервисОтвет(404);
		
		Возврат Ответ;
		
	КонецЕсли;
	
	// Вернуть данные строки.
	СтрокаДокумента = НужныйДокумент.Товары[Число(ЗапросНомерСтроки) - 1];
	
	// Преобразовать данные строки в XML.
	ЗаписьXML = Новый ЗаписьXML;
	ЗаписьXML.УстановитьСтроку();
	ЗаписьXML.ЗаписатьОбъявлениеXML();
	ЗаписьXML.ЗаписатьНачалоЭлемента("answer");
	ЗаписьXML.ЗаписатьНачалоЭлемента("СтрокаНакладной");
	ЗаписьXML.ЗаписатьАтрибут("Товар",      СтрокаДокумента.Товар.Наименование);
	ЗаписьXML.ЗаписатьАтрибут("Цена",       Строка(СтрокаДокумента.Цена));
	ЗаписьXML.ЗаписатьАтрибут("Количество", Строка(СтрокаДокумента.Количество));
	ЗаписьXML.ЗаписатьАтрибут("Сумма",      Строка(СтрокаДокумента.Сумма));
	ЗаписьXML.ЗаписатьКонецЭлемента();
	ЗаписьXML.ЗаписатьКонецЭлемента();
	XMLСтрока = ЗаписьXML.Закрыть();
	
	// Передать строку в HTTP ответ.
	Ответ = Новый HTTPСервисОтвет(200);
	Ответ.УстановитьТелоИзСтроки(XMLСтрока);
	
	Возврат Ответ;
	
КонецФункции
```

Пример функции обработчика входящего HTTP-запроса









1С-клиент 👉 PHP

HTTP



В 1С есть HTTP-клиент!

```
ЗащищенноеСоединение = Новый ЗащищенноеСоединениеOpenSSL(Неопределено, Новый СертификатыУдостоверяющихЦентровОС);
Соединение = Новый HTTPСоединение("api.github.com", 443,,,, 30, ЗащищенноеСоединение);
Запрос = Новый HTTPЗапрос("/events");
Ответ = Соединение.Получить(Запрос);
Поток = Ответ.ПолучитьТелоКакПоток();
Кодировка = "utf-8";

Ридер = Новый ЧтениеJSON;
Ридер.ОткрытьПоток(Поток, Кодировка);
Результат = ПрочитатьJSON(Ридер);
Ридер.Закрыть();
```




Что если запрос из 1С не дошел до PHP-приложения?

- **1С-Шина** https://youtu.be/FdIu9bIOJus
- **Сервисы интеграции**
- **Планы обмена** https://youtu.be/fLHDaMfJMkM

  Что изменилось с момента последнего чтения?
- Поставить задачу для команды 1С-разработчиков!
- Использовать только **PULL-модель** со стороны PHP



Идея: веб-хуки

- При появлении события в 1С не передавать пакет всех данных
- Вызывать веб-хук: `https://`**`php-project`**`/api/newInvoiceFrom1C?guid=…`
- PHP-код сходит в 1С и сам заберёт всю необходимую информацию (PULL-модель)
- **Проще поддерживать**, меньше разработки со стороны 1С-команды



Совет: упрощать синхронизацию!

- Вытягивайте справочники целиком, **PULL-модель из PHP**
- Не заморачивайтесь с отправкой только изменившихся элементов со стороны 1С (справочники)
- Вся разработка по максимуму на одной стороне (на PHP)
- Сокращайте взаимодействие и необходимость координации PHP и 1С-команд разработчиков (1С REST в помощь)



Что было в докладе, я всё проспал?

- В 1С встроено REST API
- Не хватает REST API?
  Легко написать собственный HTTP-контроллер на 1С
- Используем UUID для идентификаторов
- Начинайте с максимально простых решений (а они есть!)

```
file_get_contents('https://server1c/Catalog_Контрагенты?$expand=*');
```




Интеграция PHP-проекта с 1С





Пётр Мязин

https://5minphp.ru/2022.pdf

https://t.me/petrmyazin

